some notes of mine

hejohns' notes

hejohns

December 25, 2022

made with kaobook
set in EB Garamond

Contents

List of Figures

List of Tables

Papers/Articles

2012 - Soare, *Formalism and intuition in computability*

Tags: computability,history

Date: 2022-12-17

up to p.9

f

1981 - Kleene, *Origins of recursive function theory*

Tags: logic,computability,history

Date: December 25, 2022

need to read last 10 pages

nice review by Steward Shapiro, 1990

λ-defineable = Church, recursive = Gödel, Herbrand, computable = Turing,

(although until 193?, recursive $\mapsto$ primitive recursive, for Gödel, now recursive $\mapsto$ Herbrand-Gödel general recursive)

Book Reviews

1910,1995 – Makiko Nakano, *Makiko's Diary*

Tags:

Date: December 25, 2022

Translated by Kazuko Smith

sticky rice (desserts) = ½sticky rice + ½normal

1977,1984 – Joe Stoy, *Denotational Semantics*

Tags:

Date: December 25, 2022

Presentations/Lectures

2007 - Bryan Cantrill, *Dtrace*

Tags:

Date: December 25, 2022

Link: recording

A

Dictionary

AFAI*

Tags: acronyms

Date: 2022-12-21

As Far As I (Can Tell / Know / ...)

Craig's Trick

Tags: logic,computability

Date: 2022-12-18

Theorem (Craig's Trick)
From Mathew (MATH 684): S ce set of sentences $\implies \exists S^*.S^*$ computable $\wedge$ they have the same theory.

(my terminology) a theory is computable $\iff$ it is ce [1]

ie a theory is computably axiomatized $\iff$ it is computably enumerably axiomatized

So ce theories are just as effective as computable ones, which is good news for axiom schemas. A priori, it's not clear that theories w/ axiom schemas are as effective as finitely axiomatizable ones, but intuitively, axiom schemas are of the same character, are "easily checkable". Craig's trick formally grounds this.
eg PA is as effective as PA^-. (although PA^- has nice utility for the Entscheidungsproblem.)

1: ...sentences up to logical equivalence

Wikipedia has a similar sketch.

Proof. MATH 684: S ce, so you only have a listing of the sentences. We can make it strictly monotonic by relisting, but by adding a bunch of tautological noise or padding to each sentence (say, by conjuncting tautologies, assuming eg Gödel's prime factorization encoding) st the Gödel number is much larger. (Each sentence is relisted logically equivalently.) □

Somehow the proof itself doesn't feel intuitive, but the "intended use" of the theorem *is*.

See: Theory

Complete Lattice

Tags: order

Date: 2022-12-21

Definition (Complete Lattice)
TFAE

- lattice w/ all joins and meets $\implies$
- lattice w/ all joins, meets, top, bottom
- lattice w/ all joins [2]

2: People tend to say that the reals are "*the* complete ordered field", *in the sense that every bounded (above) set has a supremum. Which implies that every bounded (below) set has a infimum. (NOTE: "has" has a very specific meaning that everyone learns to live with.)

See: Thin Yoneda Embedding

* TODO: apparently the ambiguity of "complete" makes a difference, but afaik the following is itself correct.

Effective

Tags: logic,computability

Date: December 25, 2022

An informal notion.

I typically use "effective" to mean alt.

- "morally computable"
- "probably computable, but then I'd have to actually check"

Emotion

Tags: personal terminology

Date: December 25, 2022

I use "emotionally" similarly to "intuitively" or "morally", but "emotionally" has more to do with the gut feeling. Something that is pre-, or maybe post-, or maybe anti-, rational.

See: Informal Notion

Enumeration Operator

Tags: logic,computability

Date: 2022-12-18

Definition (Enumeration Operator)
Each enumeration reduction witness z and B determine the A, so each z determines a enumeration operator $\Phi_z : 2^\omega \to 2^\omega$.

ie $\Phi_z(B) = A \iff A \leq_e X$ witnessed by z.

$A \equiv_e B \iff A \leq_e B \wedge B \leq_e A$

Theorem

- enumeration operators compose, by inspection
- $A \subseteq B \implies \Phi(A) \subseteq \Phi(B)$ (monotonicity)
- $x \in \Phi(B) \implies \exists C.C$ finite $\wedge\, C \subseteq B \wedge x \in \Phi(C)$ (continuity)[3]

See: Scott semantics, Dana Scott's graph model of λ-calculus

From Rogers' 1967 *Theory of Recursive Functions*.

Stoy notes that the "finite approximation" requirement on Scott-continuous functions is "very close to Rogers' notion of Enumeration Reducibility." (p. 100)

An archetypical example: In Gödel's incompleteness theorems, each computable enumeration of a theory (my sense) gives rise to an enumeration of the theory (the deductive closure). In Miller's terms, there is the "deducibility operator" D that gives for each axiom set B, $D(B)$, the set of consequences.

I'm thinking of the deducibility operator the whole time.

3: which I'd call compactness

Enumeration Reducibility

From Rogers' 1967 *Theory of Recursive Functions*.

Tags: logic,computability

Date: 2022-12-18

Definition (Enumeration Reduc(tion/ible))
$A \leq_e B \iff \exists z. \forall x. x \in A \leftrightarrow \exists u. \langle x, y \rangle \in W_z \wedge D_u \subseteq B$

This definition is not as nice as (many-)one or Turing reduction, but the idea is that we want to "effectively list A using any listing (computable or not) of B". Note that enumeration reductions "only use positive information about B, and produce only positive information about A; whereas Turing reductions use and produce both positive and negative information." (paraphrased from the introduction to Russell Miller's *Non-coding Enumeration Operators*.)

where z is the Gödel code of the reduction witness, and D_u is the finite set associated with u as a canonical index (ie a tuple). [4]

4: The idea w/ u is that to list A while watching elements enter B, you should only need (to see) a finite amount of B to list a particular element $x \in A$.

Rogers' (really simple) examples:

- $\{2n | n \in \omega\} \leq_e \omega$
- A ce $\implies \forall B. A \leq_e B$

See: Enumeration Operator

Dana Scott's Graph Model

Tags: logic,λ-calculus

Date: 2022-12-18

Definition

$$\llbracket \lambda x. \rrbracket :=$$
$$\llbracket e_1 e_2 \rrbracket :=$$

See: Enumeration Operator

Herbrand's Theorem

Tags: logic

Date: 2022-12-17

TODO: convert notes from Prof. Blass' November seminar Herbrand's Theorem

Informal Notion

Tags:

Date: December 25, 2022

alt.

- a formal notion probably applies, but then I'd have to actually check
- the formal notion doesn't quite emotionally capture the concept

Example
- effective v. λ-definable, recursive, computable, ...

Knaster-Tarski

Tags: logic

Date: 2022-12-19

This theorem has many statements, and this is the easiest for me to remember. The complete lattice is often a powerset lattice.

Theorem (Knaster-Tarski Fixpoint Theorem)
Every monotone function on a complete lattice has a complete lattice of fixpoints.

Proof. Widely available. □

Example
- The deducibility operator is a monotone function on sets of sentences. [5]
  The bottom (least) fixpoint is the (deductively closed) empty theory. The top (greatest) fixpoint is the inconsistent theory, ie the set of all sentences. Consistency of the empty theory (by Gentzen's original cut elimination, or by existence of a model) says this complete lattice is nontrivial.
  Any consistent, computably axiomatizable (deductively closed) theory that proves more than the empty theory is an intermediate fixpoint– eg PA. Incompleteness says there is no intermediate fixpoint above PA that is complete, but there are at least $2^{\aleph_0}$ intermediate fixpoints above PA where we keep adding $Con(T)$ or $\neg Con(T)$. [6]

5: For simplicity, assume everything is about– and still true about– a fixed language of arithmetic.

6: Are there complete intermediate fixpoints?

See: Enumeration Operator, Dana Scott's Graph Model

Locus Solum

Tags:

Date: December 25, 2022

This is my version of Girard's dictionary.

Also afaik the most Girard paper there is

PA

Tags:

Date: December 25, 2022

Peano Arithmetic w/ induction. PA^- := PA \ induction

Realizability

Tags:

Date: December 25, 2022

This is how we can attach beamer presentations

Theory

Tags: logic

Date: 2022-12-18

An unfortunately ambiguous term, but you can usually figure it out from context, if it really matters.

I tend to use "Theory" to just mean a set of sentences, as in the Γ in the sequent $\Gamma \vdash$. So I see a finite set for "the theory of groups", and a finite set unioned w/ a schema for "Peano Arithmetic". (and the empty set for the Entscheidungsproblem.) Sometimes, people mean a deductively closed set of sentences.

TFAE

Tags: acronyms

Date: 2022-12-21

The Following Are Equivalent

When there are multiple characterizations of the same thing. Typically followed by a bullet list.

Universal Property

Tags: category theory

Date: December 25, 2022

NOTE: this is going to be in constant flux until I finally start to understand category theory...

A smattering:

afaik this is an informal notion. Things defined by universal properties should automatically be respected by lower level mappings on the base structure, that respect the lower level base structure. Except not really. [†]

[†] not at all how I want to phrase this, but I'm still trying to figure out what I mean

- fully faithful embeddings ***reflect*** limits and colimits, but don't in general *preserve* limits/colimts. [‡] [7]
  You wanted the universal property to give you preservation, but you'll have to settle w/ reflection.

7: Since terminal cones won't necessarily still be Terminal, and initial cocones won't necessarily still be initial. eg take the complete lattice with two atoms P, Q st $P \vee Q < \top$. When you take the thin Yoneda embedding, $Y(P) \vee Y(Q) = \{\perp, P\} \cup \{\perp, Q\} = \{\perp, P, Q\}$, but $Y(P \vee Q) = \{\perp, P, Q, P \vee Q\}$

TODO: how does Yoneda tie in?

Thin Yoneda Embedding

Tags: order

Date: 2022-12-21

NOTE: re name: iirc this is mentioned in Awodey in the Yoneda chapter

Theorem (see Stoy,*Denotational Semantics* 6.29. Theorem.)
Every partial order can be embedded in a complete lattice.

The partial order is often a lattice already. meets are preserved but not necessarily joins.
TODO:
A nice example from Stoy (starting ~p.89): (everything standard $\leq$ ordering) $\mathbb{Q}$ is a partial order. [§]$\mathbb{Q}$ can be completed by adding $\top = +\infty, \perp = -\infty$ and taking Dedekind cuts.

Proof. TFAE (morally)

- The presheaf category in the Yoneda embedding has all limits and colimits.
- The Yoneda embedding of a thin category *is* the powerset. TODO: how to make this more precise? How do you compose the embedding?
- Dedekind cuts.
  ie map each object to the downards closure (set of lower objects). This is a (fully faithful) embedding into the powerset lattice.

□

See: Complete Lattice, Universal Property

[‡] The Yoneda embedding preserves limits though. TODO: why? something something RAPL
[§] actually $\mathbb{Q}$ is a non-empty total order, thus a lattice, but for the theorem statement, we'll describe it this way.

Miniatures

Things I'm interested in understanding that I need to play the long con for.

apropos system T, system F, logical relations

reminder: system T is another name for the primitive recursive functionals of finite type– aka a simply typed (language w/) primitive recursive combinators. The recursion combinator at higher types extends system T to a large subset of the computable functions.

The story starts w/ Proofs and Types.

Girard presents logical relations (for strong normalization) first for STLC, then system T, then system F. As the languages get more expressive, the logical relations need to be more clever. Girard motivates the modifications at each language step in part by appealing to "logical strength considerations" (my phrase).

eg apparently strong normalization of system T $\implies$ consistency of PA, so the proof of strong normalization of system T needs to go beyond PA formalization. Apparently the original STLC proof was PA formalizable. I can't remember off the top of my head what the modification is, but iirc he appeals to the arithmetic hierarchy or something. [1] And similarly, apparently the system T proof is formalizable in second order PA, so Girard introduces reducibility candidates for the system F proof, which will probably be a long time until I understand why reducibility candidates aren't formalizable in second order PA.

1: TODO: fill this in.

To start filling in these details, a more thorough understanding of system T is required. First, all primitive recursive functions are expressible in system T by definition, but how much does recursion at higher types get you? Apparently the class of functions expressible in system T is exactly the computable functions provably total in PA.

- easy? I don't actually understand what provably total means exactly.
- hard? the forwards direction is sketched in Proofs and Types. Girard doesn't prove the backwards, instead saying that the proof for system F is more delicate and does that instead... Apparently the Avigad/Fefferman article has it though.
- ?? So how does PRA at higher types end up giving you Con(PA)?? (or rephrased, SN(T) $\implies$ Con(PA)) This seems extremely surprising from the outside. I think this is exactly the dialectica interpreation, so I'll have to actually read that first. There's a stack overflow post about this somewhere.
- koan I don't really care for primitive recursive functions and PRA much. Too low level. But is there a more intuitive way to see how PRA with higher types somehow says something about PA? Again, maybe the answer is just read the dialectica interpretation.

Related to this, I thought understanding realizability and Gödel's dialectica interpretation would clear some of the fog I have about logical relations.

Some resources: Avigad's new textbook chapter 14 (although he references earlier chapters, so I've had to read 13, parts of 8 or something, ...), Avigad/Fefferman's handbook of proof theory article, Troelstra's preface to Gödel 1958 in the collected works, Proofs and Types,

Alphabetical Index